Image captioning

Introduction

```
            Humans are creative creatures, we can easily look at an image and can describe t
hat image. What about a computer ?. Advent of deep learning made this very simple. Here we
see, how can a computer describe the image .
```

Importing necessary packages

In [1]:

```
import numpy as np
from numpy import array
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import string
import os
from PIL import Image
import glob
from pickle import dump, load
from time import time
from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import LSTM, Embedding, TimeDistributed, Dense, RepeatVector,\
                         Activation, Flatten, Reshape, concatenate, Dropout, BatchNormalization
from keras.optimizers import Adam, RMSprop
from keras.layers.wrappers import Bidirectional
from keras.layers.merge import add
from keras.applications.inception_v3 import InceptionV3
from keras.preprocessing import image
from keras.models import Model
from keras import Input, layers
from keras import optimizers
from keras.applications.inception_v3 import preprocess_input
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical
```

```
Using TensorFlow backend.
```

Summary

```
        Importing all the necessary packages.
```

Loading text document

In [2]:

```
def load_document(filename):
    file = open(filename, 'r') #opening the file
    text = file.read() # reading the all text in the file
    file.close() #closing the file
    return text # textr is returned here

filename = "text_data/Flickr8k.token.txt" # Giving the filepath
document = load_document(filename) # Document is loaded here
print(document[865:1389]) # displaying the document
```

```
1002674143_1b742ab4b8.jpg#0 A little girl covered in paint sits in front of a painted rainbow with
her hands in a bowl .
```

```
1002674143_1b742ab4b8.jpg#1 A little girl is sitting in front of a large painted rainbow .
1002674143_1b742ab4b8.jpg#2 A small girl in the grass plays with fingerpaints in front of a white
canvas with a rainbow on it .
1002674143_1b742ab4b8.jpg#3 There is a girl with pigtails sitting in front of a rainbow painting .
1002674143_1b742ab4b8.jpg#4 Young girl with pigtails painting outside in the grass .
```

Summary

```
            loading the document, and displayng the  document.
```

Visualizing the image for above text

In [3]:

```python
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np
import cv2

img = mpimg.imread('image_data/Flicker8k_Dataset/1002674143_1b742ab4b8.jpg')
plt.imshow(img, aspect='auto')
plt.show()
```

Summary

```
                Displaying the image, from the loaded document.
```

Processing the text documents

In [4]:

```python
def create_dict(document): # idea here is to make image_id as the key and description of image as
value to dictionary
    dictionary=dict() # creating empty dictionary
    for line in document.split('\n'): # Splitting the document for every new line
        token=line.split() # Splitting the line for every white space
        if len(line) < 2: # if the line length is less than 2, we are ingoring the line
            continue
        img_id,img_desc=token[0],token[1:] # 1st split part of line is taken as img_id, remaining p
art is taken as image description
        img_id=img_id.split('.')[0] #Splitting the line and taking only img_id number, neglecting t
he format
        img_desc=' '.join(img_desc) # Joining all the descriptions with white space
        if img_id not in dictionary: # if img_id is not in the dictionary, we are creating a empty
list and assigning it to the key
            dictionary[img_id]=list()
        dictionary[img_id].append(img_desc) # for every img_id as key, we are giving description as
value
    return dictionary
descriptions=create_dict(document)
print('Loaded: ',len(descriptions))
```

```
Loaded:  8092
```

Summary

```
        Here we are creating a dictionary,i.e., image id's as keys and descriptions as value
s to the keys.
```

In [5]:

```
list(descriptions.keys())[:10]
```

Out[5]:

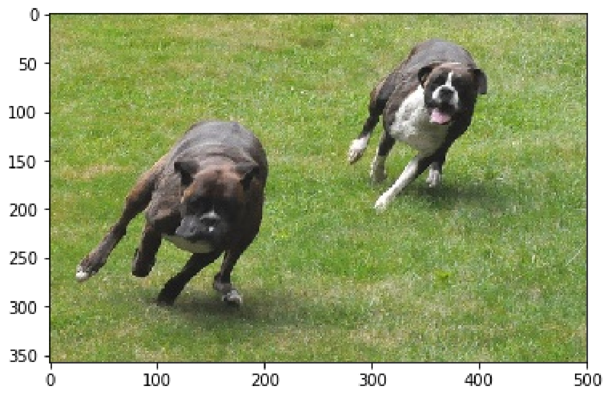
```
['3437034427_6df5e9fbf9',
 '3564312955_716e86c48b',
 '3507670136_2e5f94accf',
 '3481884992_45770ec698',
 '373219198_149af371d9',
 '3042483842_beb23828b9',
 '262570082_6364f58f33',
 '2866696346_4dcccbd3a5',
 '3598447435_f66cd10bd6',
 '3103264875_2a8d534abc']
```

Checking whether all the descriptions are correctly matched to image_id's.

In [6]:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np
import cv2

img = mpimg.imread('image_data/Flicker8k_Dataset/3583065748_7d149a865c.jpg')
plt.imshow(img, aspect='auto')
plt.show()
```



In [7]:

```
descriptions['3583065748_7d149a865c']
```

Out[7]:

```
['A black and a black dog are running .',
 'A black and white dog chases a bigger dog across the grass .',
 'One dog chases another in the grass .',
 'Two brindle dogs running in the grass .',
 'Two dogs run through the grass .']
```

In [8]:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np
import cv2

img = mpimg.imread('image_data/Flicker8k_Dataset/2528521798_fb689eba8d.jpg')
plt.imshow(img, aspect='auto')
plt.show()
```

In [9]:

```
descriptions['2528521798_fb689eba8d']
```

Out[9]:

```
['A white car racing in the dirt and water',
 'A white race car drives through a puddle .',
 'A white race car makes a splash through a wet track .',
 'A white race car splashes through a puddle on a dirt road',
 'A white rally car is throwing mud into the air as it approaches a bend in the track .']
```

Summary

```
              Checking whether the keys in the dictionary have similar values or not.
```

Data Cleaning

In [10]:

```
import string
def cleaning_text(descriptions): # here we are defining function to claen the text for the img_id'
s.
    for keys, values in descriptions.items(): # checking for evey key
        for i in range(len(values)):
            desc=values[i] # Evey description is kept into desc
            desc=desc.split() # splitting the text with white spaces
            desc=[word.lower() for word in desc] #Converting all the words to lower case
            desc=[word.translate(str.maketrans("","", string.punctuation)) for word in desc] # Remo
ving all the punctuation marks
            desc=[word for word in desc if len(word)>1] #Removing all the one letter words like 'a'
and 's'
            desc=[word for word in desc if word.isalpha()] #Removing all words with number in them
            values[i]=' '.join(desc) # Joining all the descriptions
cleaning_text(descriptions)
```

Summary

```
           Here we are cleaning the descriptions
             Steps:
                     *. Converting all the words to lower case.
                     *. Removing all the punctuations
                     *. Removing one letter word's like 'a' and 's'
                     *. Removing all words with number in them.
```

In [11]:

```
descriptions['2528521798_fb689eba8d']
```

Out[11]:

```
['white car racing in the dirt and water',
 'white race car drives through puddle',
 'white race car makes splash through wet track',
 'white race car splashes through puddle on dirt road',
 'white rally car is throwing mud into the air as it approaches bend in the track']
```

Converting all the text in the descriptions to vocabulary

In [12]:

```
def create_vocab(descriptions): #Here we are converting all the text into words
    desc_vocab=set() # creating empty set
    for key in descriptions.keys(): # we are doing this operation for evey key
        for d in descriptions[key]: # for every description
            d=d.split() # Splitting the description into words
            desc_vocab.update(d) # adding all the splitted words into the set
    return desc_vocab
vocabulary=create_vocab(descriptions)
print('Vocabulary size  :', len(vocabulary))
```

```
Vocabulary size  : 8763
```

Summary

```
            *. Creating a vocabulary set with all the words in the descriptions.
            *. There are 8763 unique words in the descriptions
```

In [13]:

```
print(list(vocabulary)[:5])
```

```
['farm', 'adolescent', 'lanterns', 'fruits', 'live']
```

Saving descriptions of text into a file

In [14]:

```
def save_description(descriptions,filename): # Here are saving all the descriptions into a file
    description_list=list() # Creating a empty list
    for key,values in descriptions.items(): # for every key
        for desc in values: # for every description
            description_list.append(key + ' ' + desc) # adding key and description
    data='\n'.join(description_list) # every description is represented in a new line
    file=open(filename,'w') # creating a file
    file.write(data) # saving the data into file
    file.close() # closing the file
save_description(descriptions, 'descriptions.txt')
```

Summary

```
                *. Saving the descriptions file,i.e., image id and description foor that image.
```

Loading the predefined train image_id's

In [15]:

```
def load_train_ids(data): # Here we are loading all the predefined trained dataset of images id's
    document=load_document(data) #reading all the image_id's into document
    train_data=list() # Creating empty list
    for line in document.split('\n'): #Splitting the document by every new line character
        if len(line)<1:
            continue
        train_image_id=line.split('.')[0] # removing the .jpeg extension
        train_data.append(train_image_id) # appending all the image_id's into a list
    return set(train_data)
train=load_train_ids('text_data/Flickr_8k.trainImages.txt')
print('Total no. of train images :', len(train))
```

```
Total no. of train images : 6000
```

Loading train images

In [16]:

```
train_images_file = 'text_data/Flickr_8k.trainImages.txt'
train_img = set(open(train_images_file, 'r').read().strip().split('\n')) #loading all the train
image id's
```

Loading test images

In [17]:

```
test_images_file = 'text_data/Flickr_8k.testImages.txt'
test_img = set(open(test_images_file, 'r').read().strip().split('\n')) # Loading all the test image
id's
```

Summary

```
            Loading all the train and test images
```

Loading descriptions for trained images from the saved descriptions file

In [18]:

```
def load_train_descriptions(filename, dataset):
    document=load_document(filename) # reading the file
    descriptions_train=dict()
    for line in document.split('\n'): # for every line
        token=line.split() # splitting line based on white space
        img_id,img_desc=token[0], token[1:]   # 1st word is taken as img_id and remaining part is ta
ken as description
        if img_id in dataset: # for every image in given sample
            if img_id not in descriptions_train:
                descriptions_train[img_id]=list()
            desc='startseq ' + ' '.join(img_desc)+ ' endseq' # adding startseq and endseq to every
description
            descriptions_train[img_id].append(desc) #appending all the keys
    return descriptions_train
train_descriptions=load_train_descriptions('descriptions.txt',train)
print('Total no. of train features are :', len(train_descriptions))
```

```
Total no. of train features are : 6000
```

In [19]:

```
list(train_descriptions.keys())[420]
```

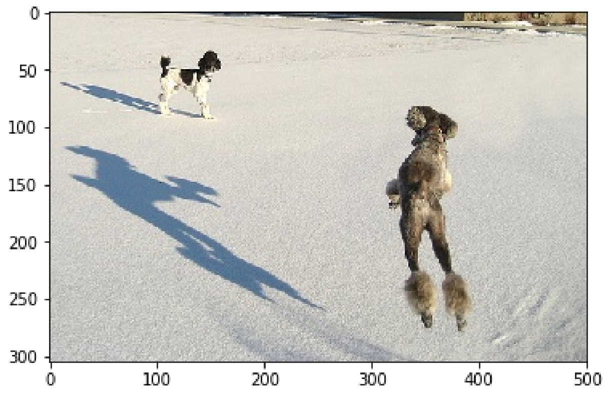
Out[19]:

```
'2676937700_456134c7b5'
```

In [20]:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np
import cv2

img = mpimg.imread('image_data/Flicker8k_Dataset/2056042552_f59e338533.jpg')
plt.imshow(img, aspect='auto')
plt.show()
```



In [21]:

```
train_descriptions['2056042552_f59e338533']
```

Out[21]:

```
['startseq grey dog is jumping toward black and white dog in the snow endseq',
 'startseq two dogs playing on the beach endseq',
 'startseq two dogs play on the beach endseq',
 'startseq two french poodles romp on snowy field endseq',
 'startseq two poodles are in the snow and one is jumping high endseq']
```

Summary

For every word, we are adding 'startseq' at the beginning of the description and 'endseq' at the end descriptions.

Loading the InceptionV3 model

In [22]:

```
model = InceptionV3(weights='imagenet') #loading the model
print(model.summary()) #print summary
```

```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, None, None, 3 0

__________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, None, None, 3 864         input_1[0][0]
__________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, None, None, 3 96          conv2d_1[0][0]
__________________________________________________________________________________________________
activation_1 (Activation)       (None, None, None, 3 0           batch_normalization_1[0][0]
__________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, None, None, 3 9216        activation_1[0][0]
__________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, None, None, 3 96          conv2d_2[0][0]
__________________________________________________________________________________________________
activation_2 (Activation)       (None, None, None, 3 0           batch_normalization_2[0][0]
__________________________________________________________________________________________________
conv2d_3 (Conv2D)               (None, None, None, 6 18432       activation_2[0][0]
__________________________________________________________________________________________________
batch_normalization_3 (BatchNor (None, None, None, 6 192         conv2d_3[0][0]
__________________________________________________________________________________________________
activation_3 (Activation)       (None, None, None, 6 0           batch_normalization_3[0][0]
__________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, None, None, 6 0           activation_3[0][0]
__________________________________________________________________________________________________
conv2d_4 (Conv2D)               (None, None, None, 8 5120        max_pooling2d_1[0][0]
__________________________________________________________________________________________________
batch_normalization_4 (BatchNor (None, None, None, 8 240         conv2d_4[0][0]
__________________________________________________________________________________________________
activation_4 (Activation)       (None, None, None, 8 0           batch_normalization_4[0][0]
__________________________________________________________________________________________________
conv2d_5 (Conv2D)               (None, None, None, 1 138240      activation_4[0][0]
__________________________________________________________________________________________________
batch_normalization_5 (BatchNor (None, None, None, 1 576         conv2d_5[0][0]
__________________________________________________________________________________________________
activation_5 (Activation)       (None, None, None, 1 0           batch_normalization_5[0][0]
__________________________________________________________________________________________________
max_pooling2d_2 (MaxPooling2D)  (None, None, None, 1 0           activation_5[0][0]
__________________________________________________________________________________________________
conv2d_9 (Conv2D)               (None, None, None, 6 12288       max_pooling2d_2[0][0]
__________________________________________________________________________________________________
batch_normalization_9 (BatchNor (None, None, None, 6 192         conv2d_9[0][0]
__________________________________________________________________________________________________
activation_9 (Activation)       (None, None, None, 6 0           batch_normalization_9[0][0]
__________________________________________________________________________________________________
conv2d_7 (Conv2D)               (None, None, None, 4 9216        max_pooling2d_2[0][0]
__________________________________________________________________________________________________
conv2d_10 (Conv2D)              (None, None, None, 9 55296       activation_9[0][0]
__________________________________________________________________________________________________
batch_normalization_7 (BatchNor (None, None, None, 4 144         conv2d_7[0][0]
__________________________________________________________________________________________________
batch_normalization_10 (BatchNo (None, None, None, 9 288         conv2d_10[0][0]
__________________________________________________________________________________________________
activation_7 (Activation)       (None, None, None, 4 0           batch_normalization_7[0][0]
__________________________________________________________________________________________________
activation_10 (Activation)      (None, None, None, 9 0           batch_normalization_10[0][0]
__________________________________________________________________________________________________
average_pooling2d_1 (AveragePoo (None, None, None, 1 0           max_pooling2d_2[0][0]
__________________________________________________________________________________________________
conv2d_6 (Conv2D)               (None, None, None, 6 12288       max_pooling2d_2[0][0]
__________________________________________________________________________________________________
conv2d_8 (Conv2D)               (None, None, None, 6 76800       activation_7[0][0]
__________________________________________________________________________________________________
conv2d_11 (Conv2D)              (None, None, None, 9 82944       activation_10[0][0]
__________________________________________________________________________________________________
conv2d_12 (Conv2D)              (None, None, None, 3 6144        average_pooling2d_1[0][0]
__________________________________________________________________________________________________
batch_normalization_6 (BatchNor (None, None, None, 6 192         conv2d_6[0][0]
__________________________________________________________________________________________________
batch_normalization_8 (BatchNor (None, None, None, 6 192         conv2d_8[0][0]
__________________________________________________________________________________________________
batch_normalization_11 (BatchNo (None, None, None, 9 288         conv2d_11[0][0]
__________________________________________________________________________________________________
batch_normalization_12 (BatchNo (None, None, None, 3 96          conv2d_12[0][0]
__________________________________________________________________________________________________
activation_6 (Activation)       (None, None, None, 6 0           batch_normalization_6[0][0]
__________________________________________________________________________________________________
activation_8 (Activation)       (None, None, None, 6 0           batch_normalization_8[0][0]
__________________________________________________________________________________________________
activation_11 (Activation)      (None, None, None, 9 0           batch_normalization_11[0][0]
__________________________________________________________________________________________________
activation_12 (Activation)      (None, None, None, 3 0           batch_normalization_12[0][0]
__________________________________________________________________________________________________
mixed0 (Concatenate)            (None, None, None, 2 0           activation_6[0][0]
                                                                 activation_8[0][0]
                                                                 activation_11[0][0]
                                                                 activation_12[0][0]
__________________________________________________________________________________________________
conv2d_16 (Conv2D)              (None, None, None, 6 16384       mixed0[0][0]
__________________________________________________________________________________________________
batch_normalization_16 (BatchNo (None, None, None, 6 192         conv2d_16[0][0]
__________________________________________________________________________________________________
activation_16 (Activation)      (None, None, None, 6 0           batch_normalization_16[0][0]
__________________________________________________________________________________________________
conv2d_14 (Conv2D)              (None, None, None, 4 12288       mixed0[0][0]
conv2d_14 (Conv2D)              (None, None, None, 4 12288       mixed0[0][0]
__________________________________________________________________________________________________
conv2d_17 (Conv2D)              (None, None, None, 9 55296       activation_16[0][0]
__________________________________________________________________________________________________
batch_normalization_14 (BatchNo (None, None, None, 4 144         conv2d_14[0][0]
__________________________________________________________________________________________________
batch_normalization_17 (BatchNo (None, None, None, 9 288         conv2d_17[0][0]
__________________________________________________________________________________________________
activation_14 (Activation)      (None, None, None, 4 0           batch_normalization_14[0][0]
__________________________________________________________________________________________________
activation_17 (Activation)      (None, None, None, 9 0           batch_normalization_17[0][0]
__________________________________________________________________________________________________
average_pooling2d_2 (AveragePoo (None, None, None, 2 0           mixed0[0][0]
__________________________________________________________________________________________________
conv2d_13 (Conv2D)              (None, None, None, 6 16384       mixed0[0][0]
__________________________________________________________________________________________________
conv2d_15 (Conv2D)              (None, None, None, 6 76800       activation_14[0][0]
__________________________________________________________________________________________________
conv2d_18 (Conv2D)              (None, None, None, 9 82944       activation_17[0][0]
__________________________________________________________________________________________________
conv2d_19 (Conv2D)              (None, None, None, 6 16384       average_pooling2d_2[0][0]
__________________________________________________________________________________________________
batch_normalization_13 (BatchNo (None, None, None, 6 192         conv2d_13[0][0]
__________________________________________________________________________________________________
batch_normalization_15 (BatchNo (None, None, None, 6 192         conv2d_15[0][0]
__________________________________________________________________________________________________
batch_normalization_18 (BatchNo (None, None, None, 9 288         conv2d_18[0][0]
__________________________________________________________________________________________________
batch_normalization_19 (BatchNo (None, None, None, 6 192         conv2d_19[0][0]
__________________________________________________________________________________________________
activation_13 (Activation)      (None, None, None, 6 0           batch_normalization_13[0][0]
__________________________________________________________________________________________________
activation_15 (Activation)      (None, None, None, 6 0           batch_normalization_15[0][0]
__________________________________________________________________________________________________
activation_18 (Activation)      (None, None, None, 9 0           batch_normalization_18[0][0]
__________________________________________________________________________________________________
activation_19 (Activation)      (None, None, None, 6 0           batch_normalization_19[0][0]
__________________________________________________________________________________________________
mixed1 (Concatenate)            (None, None, None, 2 0           activation_13[0][0]
                                                                 activation_15[0][0]
                                                                 activation_18[0][0]
                                                                 activation_19[0][0]
__________________________________________________________________________________________________
conv2d_23 (Conv2D)              (None, None, None, 6 18432       mixed1[0][0]
__________________________________________________________________________________________________
batch_normalization_23 (BatchNo (None, None, None, 6 192         conv2d_23[0][0]
__________________________________________________________________________________________________
activation_23 (Activation)      (None, None, None, 6 0           batch_normalization_23[0][0]
__________________________________________________________________________________________________
conv2d_21 (Conv2D)              (None, None, None, 4 13824       mixed1[0][0]
__________________________________________________________________________________________________
conv2d_24 (Conv2D)              (None, None, None, 9 55296       activation_23[0][0]
__________________________________________________________________________________________________
batch_normalization_21 (BatchNo (None, None, None, 4 144         conv2d_21[0][0]
__________________________________________________________________________________________________
batch_normalization_24 (BatchNo (None, None, None, 9 288         conv2d_24[0][0]
__________________________________________________________________________________________________
activation_21 (Activation)      (None, None, None, 4 0           batch_normalization_21[0][0]
__________________________________________________________________________________________________
activation_24 (Activation)      (None, None, None, 9 0           batch_normalization_24[0][0]
__________________________________________________________________________________________________
average_pooling2d_3 (AveragePoo (None, None, None, 2 0           mixed1[0][0]
__________________________________________________________________________________________________
conv2d_20 (Conv2D)              (None, None, None, 6 18432       mixed1[0][0]
__________________________________________________________________________________________________
conv2d_22 (Conv2D)              (None, None, None, 6 76800       activation_21[0][0]
__________________________________________________________________________________________________
conv2d_25 (Conv2D)              (None, None, None, 9 82944       activation_24[0][0]
__________________________________________________________________________________________________
conv2d_26 (Conv2D)              (None, None, None, 6 18432       average_pooling2d_3[0][0]
__________________________________________________________________________________________________
batch_normalization_20 (BatchNo (None, None, None, 6 192         conv2d_20[0][0]
__________________________________________________________________________________________________
batch_normalization_22 (BatchNo (None, None, None, 6 192         conv2d_22[0][0]
__________________________________________________________________________________________________
batch_normalization_25 (BatchNo (None, None, None, 9 288         conv2d_25[0][0]
__________________________________________________________________________________________________
batch_normalization_26 (BatchNo (None, None, None, 6 192         conv2d_26[0][0]
batch_normalization_26 (BatchNo (None, None, None, 6 192         conv2d_26[0][0]
__________________________________________________________________________________________________
activation_20 (Activation)      (None, None, None, 6 0           batch_normalization_20[0][0]
__________________________________________________________________________________________________
activation_22 (Activation)      (None, None, None, 6 0           batch_normalization_22[0][0]
__________________________________________________________________________________________________
activation_25 (Activation)      (None, None, None, 9 0           batch_normalization_25[0][0]
__________________________________________________________________________________________________
activation_26 (Activation)      (None, None, None, 6 0           batch_normalization_26[0][0]
__________________________________________________________________________________________________
mixed2 (Concatenate)            (None, None, None, 2 0           activation_20[0][0]
                                                                 activation_22[0][0]
                                                                 activation_25[0][0]
                                                                 activation_26[0][0]
__________________________________________________________________________________________________
conv2d_28 (Conv2D)              (None, None, None, 6 18432       mixed2[0][0]
__________________________________________________________________________________________________
batch_normalization_28 (BatchNo (None, None, None, 6 192         conv2d_28[0][0]
__________________________________________________________________________________________________
activation_28 (Activation)      (None, None, None, 6 0           batch_normalization_28[0][0]
__________________________________________________________________________________________________
conv2d_29 (Conv2D)              (None, None, None, 9 55296       activation_28[0][0]
__________________________________________________________________________________________________
batch_normalization_29 (BatchNo (None, None, None, 9 288         conv2d_29[0][0]
__________________________________________________________________________________________________
activation_29 (Activation)      (None, None, None, 9 0           batch_normalization_29[0][0]
__________________________________________________________________________________________________
conv2d_27 (Conv2D)              (None, None, None, 3 995328      mixed2[0][0]
__________________________________________________________________________________________________
conv2d_30 (Conv2D)              (None, None, None, 9 82944       activation_29[0][0]
__________________________________________________________________________________________________
batch_normalization_27 (BatchNo (None, None, None, 3 1152        conv2d_27[0][0]
__________________________________________________________________________________________________
batch_normalization_30 (BatchNo (None, None, None, 9 288         conv2d_30[0][0]
__________________________________________________________________________________________________
activation_27 (Activation)      (None, None, None, 3 0           batch_normalization_27[0][0]
__________________________________________________________________________________________________
activation_30 (Activation)      (None, None, None, 9 0           batch_normalization_30[0][0]
__________________________________________________________________________________________________
max_pooling2d_3 (MaxPooling2D)  (None, None, None, 2 0           mixed2[0][0]
__________________________________________________________________________________________________
mixed3 (Concatenate)            (None, None, None, 7 0           activation_27[0][0]
                                                                 activation_30[0][0]
                                                                 max_pooling2d_3[0][0]
__________________________________________________________________________________________________
conv2d_35 (Conv2D)              (None, None, None, 1 98304       mixed3[0][0]
__________________________________________________________________________________________________
batch_normalization_35 (BatchNo (None, None, None, 1 384         conv2d_35[0][0]
__________________________________________________________________________________________________
activation_35 (Activation)      (None, None, None, 1 0           batch_normalization_35[0][0]
__________________________________________________________________________________________________
conv2d_36 (Conv2D)              (None, None, None, 1 114688      activation_35[0][0]
__________________________________________________________________________________________________
batch_normalization_36 (BatchNo (None, None, None, 1 384         conv2d_36[0][0]
__________________________________________________________________________________________________
activation_36 (Activation)      (None, None, None, 1 0           batch_normalization_36[0][0]
__________________________________________________________________________________________________
conv2d_32 (Conv2D)              (None, None, None, 1 98304       mixed3[0][0]
__________________________________________________________________________________________________
conv2d_37 (Conv2D)              (None, None, None, 1 114688      activation_36[0][0]
__________________________________________________________________________________________________
batch_normalization_32 (BatchNo (None, None, None, 1 384         conv2d_32[0][0]
__________________________________________________________________________________________________
batch_normalization_37 (BatchNo (None, None, None, 1 384         conv2d_37[0][0]
__________________________________________________________________________________________________
activation_32 (Activation)      (None, None, None, 1 0           batch_normalization_32[0][0]
__________________________________________________________________________________________________
activation_37 (Activation)      (None, None, None, 1 0           batch_normalization_37[0][0]
__________________________________________________________________________________________________
conv2d_33 (Conv2D)              (None, None, None, 1 114688      activation_32[0][0]
__________________________________________________________________________________________________
conv2d_38 (Conv2D)              (None, None, None, 1 114688      activation_37[0][0]
__________________________________________________________________________________________________
batch_normalization_33 (BatchNo (None, None, None, 1 384         conv2d_33[0][0]
__________________________________________________________________________________________________
batch_normalization_38 (BatchNo (None, None, None, 1 384         conv2d_38[0][0]
__________________________________________________________________________________________________
activation_33 (Activation)      (None, None, None, 1 0           batch_normalization_33[0][0]
activation_33 (Activation)      (None, None, None, 1 0           batch_normalization_33[0][0]
__________________________________________________________________________________________________
activation_38 (Activation)      (None, None, None, 1 0           batch_normalization_38[0][0]
__________________________________________________________________________________________________
average_pooling2d_4 (AveragePoo (None, None, None, 7 0           mixed3[0][0]
__________________________________________________________________________________________________
conv2d_31 (Conv2D)              (None, None, None, 1 147456      mixed3[0][0]
__________________________________________________________________________________________________
conv2d_34 (Conv2D)              (None, None, None, 1 172032      activation_33[0][0]
__________________________________________________________________________________________________
conv2d_39 (Conv2D)              (None, None, None, 1 172032      activation_38[0][0]
__________________________________________________________________________________________________
conv2d_40 (Conv2D)              (None, None, None, 1 147456      average_pooling2d_4[0][0]
__________________________________________________________________________________________________
batch_normalization_31 (BatchNo (None, None, None, 1 576         conv2d_31[0][0]
__________________________________________________________________________________________________
batch_normalization_34 (BatchNo (None, None, None, 1 576         conv2d_34[0][0]
__________________________________________________________________________________________________
batch_normalization_39 (BatchNo (None, None, None, 1 576         conv2d_39[0][0]
__________________________________________________________________________________________________
batch_normalization_40 (BatchNo (None, None, None, 1 576         conv2d_40[0][0]
__________________________________________________________________________________________________
activation_31 (Activation)      (None, None, None, 1 0           batch_normalization_31[0][0]
__________________________________________________________________________________________________
activation_34 (Activation)      (None, None, None, 1 0           batch_normalization_34[0][0]
__________________________________________________________________________________________________
activation_39 (Activation)      (None, None, None, 1 0           batch_normalization_39[0][0]
__________________________________________________________________________________________________
activation_40 (Activation)      (None, None, None, 1 0           batch_normalization_40[0][0]
__________________________________________________________________________________________________
mixed4 (Concatenate)            (None, None, None, 7 0           activation_31[0][0]
                                                                 activation_34[0][0]
                                                                 activation_39[0][0]
                                                                 activation_40[0][0]
__________________________________________________________________________________________________
conv2d_45 (Conv2D)              (None, None, None, 1 122880      mixed4[0][0]
__________________________________________________________________________________________________
batch_normalization_45 (BatchNo (None, None, None, 1 480         conv2d_45[0][0]
__________________________________________________________________________________________________
activation_45 (Activation)      (None, None, None, 1 0           batch_normalization_45[0][0]
__________________________________________________________________________________________________
conv2d_46 (Conv2D)              (None, None, None, 1 179200      activation_45[0][0]
__________________________________________________________________________________________________
batch_normalization_46 (BatchNo (None, None, None, 1 480         conv2d_46[0][0]
__________________________________________________________________________________________________
activation_46 (Activation)      (None, None, None, 1 0           batch_normalization_46[0][0]
__________________________________________________________________________________________________
conv2d_42 (Conv2D)              (None, None, None, 1 122880      mixed4[0][0]
__________________________________________________________________________________________________
conv2d_47 (Conv2D)              (None, None, None, 1 179200      activation_46[0][0]
__________________________________________________________________________________________________
batch_normalization_42 (BatchNo (None, None, None, 1 480         conv2d_42[0][0]
__________________________________________________________________________________________________
batch_normalization_47 (BatchNo (None, None, None, 1 480         conv2d_47[0][0]
__________________________________________________________________________________________________
activation_42 (Activation)      (None, None, None, 1 0           batch_normalization_42[0][0]
__________________________________________________________________________________________________
activation_47 (Activation)      (None, None, None, 1 0           batch_normalization_47[0][0]
__________________________________________________________________________________________________
conv2d_43 (Conv2D)              (None, None, None, 1 179200      activation_42[0][0]
__________________________________________________________________________________________________
conv2d_48 (Conv2D)              (None, None, None, 1 179200      activation_47[0][0]
__________________________________________________________________________________________________
batch_normalization_43 (BatchNo (None, None, None, 1 480         conv2d_43[0][0]
__________________________________________________________________________________________________
batch_normalization_48 (BatchNo (None, None, None, 1 480         conv2d_48[0][0]
__________________________________________________________________________________________________
activation_43 (Activation)      (None, None, None, 1 0           batch_normalization_43[0][0]
__________________________________________________________________________________________________
activation_48 (Activation)      (None, None, None, 1 0           batch_normalization_48[0][0]
__________________________________________________________________________________________________
average_pooling2d_5 (AveragePoo (None, None, None, 7 0           mixed4[0][0]
__________________________________________________________________________________________________
conv2d_41 (Conv2D)              (None, None, None, 1 147456      mixed4[0][0]
__________________________________________________________________________________________________
conv2d_44 (Conv2D)              (None, None, None, 1 215040      activation_43[0][0]
__________________________________________________________________________________________________
conv2d_49 (Conv2D)              (None, None, None, 1 215040      activation_48[0][0]
conv2d_49 (Conv2D)              (None, None, None, 1 215040      activation_48[0][0]
__________________________________________________________________________________________________
conv2d_50 (Conv2D)              (None, None, None, 1 147456      average_pooling2d_5[0][0]
__________________________________________________________________________________________________
batch_normalization_41 (BatchNo (None, None, None, 1 576         conv2d_41[0][0]
__________________________________________________________________________________________________
batch_normalization_44 (BatchNo (None, None, None, 1 576         conv2d_44[0][0]
__________________________________________________________________________________________________
batch_normalization_49 (BatchNo (None, None, None, 1 576         conv2d_49[0][0]
__________________________________________________________________________________________________
batch_normalization_50 (BatchNo (None, None, None, 1 576         conv2d_50[0][0]
__________________________________________________________________________________________________
activation_41 (Activation)      (None, None, None, 1 0           batch_normalization_41[0][0]
__________________________________________________________________________________________________
activation_44 (Activation)      (None, None, None, 1 0           batch_normalization_44[0][0]
__________________________________________________________________________________________________
activation_49 (Activation)      (None, None, None, 1 0           batch_normalization_49[0][0]
__________________________________________________________________________________________________
activation_50 (Activation)      (None, None, None, 1 0           batch_normalization_50[0][0]
__________________________________________________________________________________________________
mixed5 (Concatenate)            (None, None, None, 7 0           activation_41[0][0]
                                                                 activation_44[0][0]
                                                                 activation_49[0][0]
                                                                 activation_50[0][0]
__________________________________________________________________________________________________
conv2d_55 (Conv2D)              (None, None, None, 1 122880      mixed5[0][0]
__________________________________________________________________________________________________
batch_normalization_55 (BatchNo (None, None, None, 1 480         conv2d_55[0][0]
__________________________________________________________________________________________________
activation_55 (Activation)      (None, None, None, 1 0           batch_normalization_55[0][0]
__________________________________________________________________________________________________
conv2d_56 (Conv2D)              (None, None, None, 1 179200      activation_55[0][0]
__________________________________________________________________________________________________
batch_normalization_56 (BatchNo (None, None, None, 1 480         conv2d_56[0][0]
__________________________________________________________________________________________________
activation_56 (Activation)      (None, None, None, 1 0           batch_normalization_56[0][0]
__________________________________________________________________________________________________
conv2d_52 (Conv2D)              (None, None, None, 1 122880      mixed5[0][0]
__________________________________________________________________________________________________
conv2d_57 (Conv2D)              (None, None, None, 1 179200      activation_56[0][0]
__________________________________________________________________________________________________
batch_normalization_52 (BatchNo (None, None, None, 1 480         conv2d_52[0][0]
__________________________________________________________________________________________________
batch_normalization_57 (BatchNo (None, None, None, 1 480         conv2d_57[0][0]
__________________________________________________________________________________________________
activation_52 (Activation)      (None, None, None, 1 0           batch_normalization_52[0][0]
__________________________________________________________________________________________________
activation_57 (Activation)      (None, None, None, 1 0           batch_normalization_57[0][0]
__________________________________________________________________________________________________
conv2d_53 (Conv2D)              (None, None, None, 1 179200      activation_52[0][0]
__________________________________________________________________________________________________
conv2d_58 (Conv2D)              (None, None, None, 1 179200      activation_57[0][0]
__________________________________________________________________________________________________
batch_normalization_53 (BatchNo (None, None, None, 1 480         conv2d_53[0][0]
__________________________________________________________________________________________________
batch_normalization_58 (BatchNo (None, None, None, 1 480         conv2d_58[0][0]
__________________________________________________________________________________________________
activation_53 (Activation)      (None, None, None, 1 0           batch_normalization_53[0][0]
__________________________________________________________________________________________________
activation_58 (Activation)      (None, None, None, 1 0           batch_normalization_58[0][0]
__________________________________________________________________________________________________
average_pooling2d_6 (AveragePoo (None, None, None, 7 0           mixed5[0][0]
__________________________________________________________________________________________________
conv2d_51 (Conv2D)              (None, None, None, 1 147456      mixed5[0][0]
__________________________________________________________________________________________________
conv2d_54 (Conv2D)              (None, None, None, 1 215040      activation_53[0][0]
__________________________________________________________________________________________________
conv2d_59 (Conv2D)              (None, None, None, 1 215040      activation_58[0][0]
__________________________________________________________________________________________________
conv2d_60 (Conv2D)              (None, None, None, 1 147456      average_pooling2d_6[0][0]
__________________________________________________________________________________________________
batch_normalization_51 (BatchNo (None, None, None, 1 576         conv2d_51[0][0]
__________________________________________________________________________________________________
batch_normalization_54 (BatchNo (None, None, None, 1 576         conv2d_54[0][0]
__________________________________________________________________________________________________
batch_normalization_59 (BatchNo (None, None, None, 1 576         conv2d_59[0][0]
__________________________________________________________________________________________________
batch_normalization_60 (BatchNo (None, None, None, 1 576         conv2d_60[0][0]
batch_normalization_60 (BatchNo (None, None, None, 1 576         conv2d_60[0][0]
__________________________________________________________________________________________________
activation_51 (Activation)      (None, None, None, 1 0           batch_normalization_51[0][0]
__________________________________________________________________________________________________
activation_54 (Activation)      (None, None, None, 1 0           batch_normalization_54[0][0]
__________________________________________________________________________________________________
activation_59 (Activation)      (None, None, None, 1 0           batch_normalization_59[0][0]
__________________________________________________________________________________________________
activation_60 (Activation)      (None, None, None, 1 0           batch_normalization_60[0][0]
__________________________________________________________________________________________________
mixed6 (Concatenate)            (None, None, None, 7 0           activation_51[0][0]
                                                                 activation_54[0][0]
                                                                 activation_59[0][0]
                                                                 activation_60[0][0]
__________________________________________________________________________________________________
conv2d_65 (Conv2D)              (None, None, None, 1 147456      mixed6[0][0]
__________________________________________________________________________________________________
batch_normalization_65 (BatchNo (None, None, None, 1 576         conv2d_65[0][0]
__________________________________________________________________________________________________
activation_65 (Activation)      (None, None, None, 1 0           batch_normalization_65[0][0]
__________________________________________________________________________________________________
conv2d_66 (Conv2D)              (None, None, None, 1 258048      activation_65[0][0]
__________________________________________________________________________________________________
batch_normalization_66 (BatchNo (None, None, None, 1 576         conv2d_66[0][0]
__________________________________________________________________________________________________
activation_66 (Activation)      (None, None, None, 1 0           batch_normalization_66[0][0]
__________________________________________________________________________________________________
conv2d_62 (Conv2D)              (None, None, None, 1 147456      mixed6[0][0]
__________________________________________________________________________________________________
conv2d_67 (Conv2D)              (None, None, None, 1 258048      activation_66[0][0]
__________________________________________________________________________________________________
batch_normalization_62 (BatchNo (None, None, None, 1 576         conv2d_62[0][0]
__________________________________________________________________________________________________
batch_normalization_67 (BatchNo (None, None, None, 1 576         conv2d_67[0][0]
__________________________________________________________________________________________________
activation_62 (Activation)      (None, None, None, 1 0           batch_normalization_62[0][0]
__________________________________________________________________________________________________
activation_67 (Activation)      (None, None, None, 1 0           batch_normalization_67[0][0]
__________________________________________________________________________________________________
conv2d_63 (Conv2D)              (None, None, None, 1 258048      activation_62[0][0]
__________________________________________________________________________________________________
conv2d_68 (Conv2D)              (None, None, None, 1 258048      activation_67[0][0]
__________________________________________________________________________________________________
batch_normalization_63 (BatchNo (None, None, None, 1 576         conv2d_63[0][0]
__________________________________________________________________________________________________
batch_normalization_68 (BatchNo (None, None, None, 1 576         conv2d_68[0][0]
__________________________________________________________________________________________________
activation_63 (Activation)      (None, None, None, 1 0           batch_normalization_63[0][0]
__________________________________________________________________________________________________
activation_68 (Activation)      (None, None, None, 1 0           batch_normalization_68[0][0]
__________________________________________________________________________________________________
average_pooling2d_7 (AveragePoo (None, None, None, 7 0           mixed6[0][0]
__________________________________________________________________________________________________
conv2d_61 (Conv2D)              (None, None, None, 1 147456      mixed6[0][0]
__________________________________________________________________________________________________
conv2d_64 (Conv2D)              (None, None, None, 1 258048      activation_63[0][0]
__________________________________________________________________________________________________
conv2d_69 (Conv2D)              (None, None, None, 1 258048      activation_68[0][0]
__________________________________________________________________________________________________
conv2d_70 (Conv2D)              (None, None, None, 1 147456      average_pooling2d_7[0][0]
__________________________________________________________________________________________________
batch_normalization_61 (BatchNo (None, None, None, 1 576         conv2d_61[0][0]
__________________________________________________________________________________________________
batch_normalization_64 (BatchNo (None, None, None, 1 576         conv2d_64[0][0]
__________________________________________________________________________________________________
batch_normalization_69 (BatchNo (None, None, None, 1 576         conv2d_69[0][0]
__________________________________________________________________________________________________
batch_normalization_70 (BatchNo (None, None, None, 1 576         conv2d_70[0][0]
__________________________________________________________________________________________________
activation_61 (Activation)      (None, None, None, 1 0           batch_normalization_61[0][0]
__________________________________________________________________________________________________
activation_64 (Activation)      (None, None, None, 1 0           batch_normalization_64[0][0]
__________________________________________________________________________________________________
activation_69 (Activation)      (None, None, None, 1 0           batch_normalization_69[0][0]
__________________________________________________________________________________________________
activation_70 (Activation)      (None, None, None, 1 0           batch_normalization_70[0][0]
__________________________________________________________________________________________________
mixed7 (Concatenate)            (None, None, None, 7 0           activation_61[0][0]
mixed7 (Concatenate)            (None, None, None, 7 0           activation_61[0][0]
                                                                 activation_64[0][0]
                                                                 activation_69[0][0]
                                                                 activation_70[0][0]
__________________________________________________________________________________________________
conv2d_73 (Conv2D)              (None, None, None, 1 147456      mixed7[0][0]
__________________________________________________________________________________________________
batch_normalization_73 (BatchNo (None, None, None, 1 576         conv2d_73[0][0]
__________________________________________________________________________________________________
activation_73 (Activation)      (None, None, None, 1 0           batch_normalization_73[0][0]
__________________________________________________________________________________________________
conv2d_74 (Conv2D)              (None, None, None, 1 258048      activation_73[0][0]
__________________________________________________________________________________________________
batch_normalization_74 (BatchNo (None, None, None, 1 576         conv2d_74[0][0]
__________________________________________________________________________________________________
activation_74 (Activation)      (None, None, None, 1 0           batch_normalization_74[0][0]
__________________________________________________________________________________________________
conv2d_71 (Conv2D)              (None, None, None, 1 147456      mixed7[0][0]
__________________________________________________________________________________________________
conv2d_75 (Conv2D)              (None, None, None, 1 258048      activation_74[0][0]
__________________________________________________________________________________________________
batch_normalization_71 (BatchNo (None, None, None, 1 576         conv2d_71[0][0]
__________________________________________________________________________________________________
batch_normalization_75 (BatchNo (None, None, None, 1 576         conv2d_75[0][0]
__________________________________________________________________________________________________
activation_71 (Activation)      (None, None, None, 1 0           batch_normalization_71[0][0]
__________________________________________________________________________________________________
activation_75 (Activation)      (None, None, None, 1 0           batch_normalization_75[0][0]
__________________________________________________________________________________________________
conv2d_72 (Conv2D)              (None, None, None, 3 552960      activation_71[0][0]
__________________________________________________________________________________________________
conv2d_76 (Conv2D)              (None, None, None, 1 331776      activation_75[0][0]
__________________________________________________________________________________________________
batch_normalization_72 (BatchNo (None, None, None, 3 960         conv2d_72[0][0]
__________________________________________________________________________________________________
batch_normalization_76 (BatchNo (None, None, None, 1 576         conv2d_76[0][0]
__________________________________________________________________________________________________
activation_72 (Activation)      (None, None, None, 3 0           batch_normalization_72[0][0]
__________________________________________________________________________________________________
activation_76 (Activation)      (None, None, None, 1 0           batch_normalization_76[0][0]
__________________________________________________________________________________________________
max_pooling2d_4 (MaxPooling2D)  (None, None, None, 7 0           mixed7[0][0]
__________________________________________________________________________________________________
mixed8 (Concatenate)            (None, None, None, 1 0           activation_72[0][0]
                                                                 activation_76[0][0]
                                                                 max_pooling2d_4[0][0]
__________________________________________________________________________________________________
conv2d_81 (Conv2D)              (None, None, None, 4 573440      mixed8[0][0]
__________________________________________________________________________________________________
batch_normalization_81 (BatchNo (None, None, None, 4 1344        conv2d_81[0][0]
__________________________________________________________________________________________________
activation_81 (Activation)      (None, None, None, 4 0           batch_normalization_81[0][0]
__________________________________________________________________________________________________
conv2d_78 (Conv2D)              (None, None, None, 3 491520      mixed8[0][0]
__________________________________________________________________________________________________
conv2d_82 (Conv2D)              (None, None, None, 3 1548288     activation_81[0][0]
__________________________________________________________________________________________________
batch_normalization_78 (BatchNo (None, None, None, 3 1152        conv2d_78[0][0]
__________________________________________________________________________________________________
batch_normalization_82 (BatchNo (None, None, None, 3 1152        conv2d_82[0][0]
__________________________________________________________________________________________________
activation_78 (Activation)      (None, None, None, 3 0           batch_normalization_78[0][0]
__________________________________________________________________________________________________
activation_82 (Activation)      (None, None, None, 3 0           batch_normalization_82[0][0]
__________________________________________________________________________________________________
conv2d_79 (Conv2D)              (None, None, None, 3 442368      activation_78[0][0]
__________________________________________________________________________________________________
conv2d_80 (Conv2D)              (None, None, None, 3 442368      activation_78[0][0]
__________________________________________________________________________________________________
conv2d_83 (Conv2D)              (None, None, None, 3 442368      activation_82[0][0]
__________________________________________________________________________________________________
conv2d_84 (Conv2D)              (None, None, None, 3 442368      activation_82[0][0]
__________________________________________________________________________________________________
average_pooling2d_8 (AveragePoo (None, None, None, 1 0           mixed8[0][0]
__________________________________________________________________________________________________
conv2d_77 (Conv2D)              (None, None, None, 3 409600      mixed8[0][0]
__________________________________________________________________________________________________
```

```
batch_normalization_79 (BatchNo (None, None, None, 3 1152        conv2d_79[0][0]
__________________________________________________________________________________________________
batch_normalization_80 (BatchNo (None, None, None, 3 1152        conv2d_80[0][0]
__________________________________________________________________________________________________
batch_normalization_83 (BatchNo (None, None, None, 3 1152        conv2d_83[0][0]
__________________________________________________________________________________________________
batch_normalization_84 (BatchNo (None, None, None, 3 1152        conv2d_84[0][0]
__________________________________________________________________________________________________
conv2d_85 (Conv2D)              (None, None, None, 1 245760      average_pooling2d_8[0][0]
__________________________________________________________________________________________________
batch_normalization_77 (BatchNo (None, None, None, 3 960         conv2d_77[0][0]
__________________________________________________________________________________________________
activation_79 (Activation)      (None, None, None, 3 0           batch_normalization_79[0][0]
__________________________________________________________________________________________________
activation_80 (Activation)      (None, None, None, 3 0           batch_normalization_80[0][0]
__________________________________________________________________________________________________
activation_83 (Activation)      (None, None, None, 3 0           batch_normalization_83[0][0]
__________________________________________________________________________________________________
activation_84 (Activation)      (None, None, None, 3 0           batch_normalization_84[0][0]
__________________________________________________________________________________________________
batch_normalization_85 (BatchNo (None, None, None, 1 576         conv2d_85[0][0]
__________________________________________________________________________________________________
activation_77 (Activation)      (None, None, None, 3 0           batch_normalization_77[0][0]
__________________________________________________________________________________________________
mixed9_0 (Concatenate)          (None, None, None, 7 0           activation_79[0][0]
                                                                 activation_80[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, None, None, 7 0           activation_83[0][0]
                                                                 activation_84[0][0]
__________________________________________________________________________________________________
activation_85 (Activation)      (None, None, None, 1 0           batch_normalization_85[0][0]
__________________________________________________________________________________________________
mixed9 (Concatenate)            (None, None, None, 2 0           activation_77[0][0]
                                                                 mixed9_0[0][0]
                                                                 concatenate_1[0][0]
                                                                 activation_85[0][0]
__________________________________________________________________________________________________
conv2d_90 (Conv2D)              (None, None, None, 4 917504      mixed9[0][0]
__________________________________________________________________________________________________
batch_normalization_90 (BatchNo (None, None, None, 4 1344        conv2d_90[0][0]
__________________________________________________________________________________________________
activation_90 (Activation)      (None, None, None, 4 0           batch_normalization_90[0][0]
__________________________________________________________________________________________________
conv2d_87 (Conv2D)              (None, None, None, 3 786432      mixed9[0][0]
__________________________________________________________________________________________________
conv2d_91 (Conv2D)              (None, None, None, 3 1548288     activation_90[0][0]
__________________________________________________________________________________________________
batch_normalization_87 (BatchNo (None, None, None, 3 1152        conv2d_87[0][0]
__________________________________________________________________________________________________
batch_normalization_91 (BatchNo (None, None, None, 3 1152        conv2d_91[0][0]
__________________________________________________________________________________________________
activation_87 (Activation)      (None, None, None, 3 0           batch_normalization_87[0][0]
__________________________________________________________________________________________________
activation_91 (Activation)      (None, None, None, 3 0           batch_normalization_91[0][0]
__________________________________________________________________________________________________
conv2d_88 (Conv2D)              (None, None, None, 3 442368      activation_87[0][0]
__________________________________________________________________________________________________
conv2d_89 (Conv2D)              (None, None, None, 3 442368      activation_87[0][0]
__________________________________________________________________________________________________
conv2d_92 (Conv2D)              (None, None, None, 3 442368      activation_91[0][0]
__________________________________________________________________________________________________
conv2d_93 (Conv2D)              (None, None, None, 3 442368      activation_91[0][0]
__________________________________________________________________________________________________
average_pooling2d_9 (AveragePoo (None, None, None, 2 0           mixed9[0][0]
__________________________________________________________________________________________________
conv2d_86 (Conv2D)              (None, None, None, 3 655360      mixed9[0][0]
__________________________________________________________________________________________________
batch_normalization_88 (BatchNo (None, None, None, 3 1152        conv2d_88[0][0]
__________________________________________________________________________________________________
batch_normalization_89 (BatchNo (None, None, None, 3 1152        conv2d_89[0][0]
__________________________________________________________________________________________________
batch_normalization_92 (BatchNo (None, None, None, 3 1152        conv2d_92[0][0]
__________________________________________________________________________________________________
batch_normalization_93 (BatchNo (None, None, None, 3 1152        conv2d_93[0][0]
__________________________________________________________________________________________________
conv2d_94 (Conv2D)              (None, None, None, 1 393216      average_pooling2d_9[0][0]
__________________________________________________________________________________________________
```

```
batch_normalization_86 (BatchNo (None, None, None, 3 960         conv2d_86[0][0]
__________________________________________________________________________________________________
activation_88 (Activation)      (None, None, None, 3 0           batch_normalization_88[0][0]
__________________________________________________________________________________________________
activation_89 (Activation)      (None, None, None, 3 0           batch_normalization_89[0][0]
__________________________________________________________________________________________________
activation_92 (Activation)      (None, None, None, 3 0           batch_normalization_92[0][0]
__________________________________________________________________________________________________
activation_93 (Activation)      (None, None, None, 3 0           batch_normalization_93[0][0]
__________________________________________________________________________________________________
batch_normalization_94 (BatchNo (None, None, None, 1 576         conv2d_94[0][0]
__________________________________________________________________________________________________
activation_86 (Activation)      (None, None, None, 3 0           batch_normalization_86[0][0]
__________________________________________________________________________________________________
mixed9_1 (Concatenate)          (None, None, None, 7 0           activation_88[0][0]
                                                                 activation_89[0][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, None, None, 7 0           activation_92[0][0]
                                                                 activation_93[0][0]
__________________________________________________________________________________________________
activation_94 (Activation)      (None, None, None, 1 0           batch_normalization_94[0][0]
__________________________________________________________________________________________________
mixed10 (Concatenate)           (None, None, None, 2 0           activation_86[0][0]
                                                                 mixed9_1[0][0]
                                                                 concatenate_2[0][0]
                                                                 activation_94[0][0]
__________________________________________________________________________________________________
avg_pool (GlobalAveragePooling2 (None, 2048)         0           mixed10[0][0]
__________________________________________________________________________________________________
predictions (Dense)             (None, 1000)         2049000     avg_pool[0][0]
==================================================================================================
Total params: 23,851,784
Trainable params: 23,817,352
Non-trainable params: 34,432
__________________________________________________________________________________________________
None
```

Summary

```
        Loading the predefined InceptionV3 model .Here we are using transfer learning strate
gy .
```

Creating new model with bottleneck features

In [23]:

```
model_bottleneck=Model(model.input,model.layers[-2].output) #creating a new model with removing
last layer in the model
```

Summary

```
        Here we are creating a new model by removing last layer in the model
```

Preprocess the input image

In [24]:

```
def preprocess(image_path): # Here we aare preprocessing the image
    img=image.load_img('image_data/Flicker8k_Dataset/'+image_path,target_size=(299,299)) #Setting t
he image size to(224,224)
    x=image.img_to_array(img) #converting the image to numpy array
    x=np.expand_dims(x,axis=0) #adding one more dimension
    x=preprocess_input(x)
    return x
```

Encoding the image

In [25]:

```
def encode(image): # Here we are encoding the image as of our requirement
    image=preprocess(image) # Preprocessing the image
    fea_vec=model_bottleneck.predict(image) # Getting the encoded image feature vector from the
given image
    fea_vec=np.reshape(fea_vec, fea_vec.shape[1]) # reshaping the image from (1,2048) to (2048,)
    return fea_vec
```

Encoding train images

In [28]:

```
encoded_train_img=dict() #creating a dictionary
for img in train_img: #for evey image
    image_id = img.split('.')[0] # Removing '.jpg' extension
    encoded_train_img[image_id]=encode(img) # for every image_id as key, extracted vector of size
2048 is stored as value
```

Summary

```
Encoding the image
Steps
    *. Converting the image to numpy array
    *. With the created new model, we are predicting the image.
    *. Reshaping the image
```

In [29]:

```
import pickle
with open("image_data/encoded_train_images.pkl", "wb") as encoded_pickle: #creating a file
    pickle.dump(encoded_train_img, encoded_pickle) #storing the file as pickle file
```

Summary

```
Saving the data of encoded train images in a pickle file.
```

Encoding test images

In [30]:

```
encoded_test_img=dict() #creating a dictionary
for img in test_img: #for evey test image
    image_id = img.split('.')[0] #removing '.jpg' extension
    encoded_test_img[image_id]=encode(img) # for every image_id as key,extracted vectorof size 204
is stored as value
```

In [31]:

```
import pickle
with open("image_data/encoded_test_images.pkl", "wb") as encoded_pickle: #creating a file
    pickle.dump(encoded_test_img, encoded_pickle)# storing the file as pikle file
```

Summary

```
Saving the data of encoded test images in a pickle file.
```

Loading the trained features

In [26]:

```
train_features=load(open('image_data/encoded_train_images.pkl','rb')) #loading the stored file
print('Total no. of train features are :', len(train_features))
```

```
Total no. of train features are : 6000
```

In [27]:

```
train_captions=[] # creating a list
for key,values in train_descriptions.items(): #for every image_id
    for val in values: # for every description
        train_captions.append(val) # appending all the descriptions into a list
len(train_captions) # we will be 30000 descriptions
```

Out[27]:

```
30000
```

Summary

```
*. Here we are loading all the descriptions into a list.
*. There are 30000 captions(6000 images * 5 descriptions)
```

Counting words which occur more times

In [28]:

```
word_count_threshold=10
word_count={}
for sent in train_captions: #for every caption
    for word in sent.split(' '): #for every word split by white space
        word_count[word]=word_count.get(word,0)+1 # counting occurence of each word
vocab=[word for word in word_count if word_count[word] >= word_count_threshold] #checking the
condition
print('preprocessed words %d -> %d' % (len(word_count), len(vocab)))
```

```
preprocessed words 7578 -> 1651
```

Summary

```
        From the vocabulary of words, we are choosing only the words, which occur more th
an 10 times in the vocabulary
```

Indexing words

In [29]:

```
index_to_word={} # Creating a empty dictionary
word_to_index={} # Creating a empty dictionary
index=1
for word in vocab: #For every word in vocabulary
    index_to_word[index]=word #For every index , we are assigning a word
    word_to_index[word]=index # For every word, we are assigning an index
    index+=1
```

Summary

```
*. We are assigning a index to every word
*. We are assigning a word to every index
```

Finding the maximum length of descriptions

In [30]:

```
def lines(descriptions):
    desc=list()   # Creating a empty list
    for key in descriptions.keys(): # for every image id
        for description in descriptions[key]: # For every description for that image
            desc.append(description) # Loading all the desciptions to desc
    return desc
def maximum_length(descriptions):
    line=lines(descriptions) #loading the whole descriptions
    return max(len(d.split()) for d in line) # Splitting the words and returning the maximum length
of descriptions
max_len= maximum_length(descriptions)
print("max length of words are :",max_len)
```

```
max length of words are : 32
```

Summary

```
               Finding the max length of the descriptions for given train images.
```

Generating all the input data

In [31]:

```
def data_generator(descriptions,photos, word_to_index,max_length,num_photos_per_batch):
    X1,X2,y=list(),list(),list()
    n=0
    for key,desc in descriptions.items(): #For every image and description
        n+=1
        photo=photos[key] #loading the image id
        for d in desc:
            seq = [word_to_index[word] for word in d.split(' ') if word in word_to_index] #
Converting the word in descriptions to indexes
            for i in range(1,len(seq)): # for range of length of words in descriptions
                in_seq,out_seq=seq[:i],seq[i] #Taking word as input and next occuring word as outpu
t
                in_seq=pad_sequences([in_seq],maxlen=max_len)[0] #padding the word upto maximum len
gth of descriptions
                out_seq=to_categorical([out_seq], num_classes=vocabulary_size)[0] #Converting the t
arget value to a categorical value
                X1.append(photo) #appending the encoded image feature
                X2.append(in_seq) # Appending the input sequence of word
                y.append(out_seq) # Appending the output sequence(next occuring word)
        if n==num_photos_per_batch:
                yield [[array(X1), array(X2)], array(y)]
                X1, X2, y = list(), list(), list()
                n=0
```

Summary

```
           Generating input data
           Steps
                   *. We are taking every description
                   *. Taking every word with padding upto maximum length of description and tak
   ing next occurring word as target word.
                   *. Instead of words, we are taking index of the words
                   * Coverting all the target values to the categorical values
```

Loading GlOVE model

In [32]:

```
glove_dir='GLOVE'
```

```
glove_vectors={} #Creating a empty dictionary
f=open(os.path.join(glove_dir,'glove.6B.200d.txt'),encoding='utf-8') #loading the glove directory
for file_line in f:  # for every line in glove file
    line=file_line.split() #Splitting the line
    word=line[0] # First index of that line is word
    coefs = np.asarray(line[1:], dtype='float32') # remaining index are coefficients for that word
    glove_vectors[word] = coefs #in the dictionary, we are storing the word as key and coefficient
of word as value for that word
f.close()
print(' No. of glove vector are ' , len(glove_vectors))
```

```
 No. of glove vector are  400000
```

Summary

Loading all the vectors for each word.

In [33]:

```
vocabulary_size=(len(word_to_index))+1
vocabulary_size
```

Out[33]:

```
1652
```

In [34]:

```
dim_vector=200 #Taking only 200 dimensional vector
word_matrix=np.zeros((vocabulary_size, dim_vector)) #creating a matrix with all zeros
for word,index in word_to_index.items(): #for every word
    glove_vec=glove_vectors.get(word) # for every word, we are loading the coefficient of that
word
    if glove_vec is not None:
        word_matrix[index]=glove_vec
```

Summary

For every word, we are just taking the 200 dimensions only.

In [35]:

```
word_matrix
```

Out[35]:

```
array([[ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.16711   ,  0.016552  , -0.56985998, ...,  0.57885998,
        -0.25709   ,  0.45908001],
       [-0.0029923 , -0.79718   ,  0.0061861 , ...,  0.78188002,
        -0.52678001,  0.34862   ],
       ...,
       [ 0.16773   ,  0.26133999, -0.41501001, ...,  0.10062   ,
        -0.010848  ,  0.11345   ],
       [ 0.35479   , -0.32782999, -1.06869996, ...,  0.070415  ,
        -0.097698  , -0.41589999],
       [-0.26923999,  0.025036  , -0.84415001, ...,  0.10625   ,
        -0.30094999,  0.101     ]])
```

In [36]:

```
word_matrix.shape
```

Out[36]:

```
(1652, 200)
```

In [37]:

```
inputs1 = Input(shape=(2048,))
fe1 = Dropout(0.5)(inputs1)
fe2 = Dense(256, activation='relu')(fe1) #activation unit a relu
inputs2 = Input(shape=(max_len,))
se1 = Embedding(vocabulary_size,dim_vector, mask_zero=True)(inputs2)
se2 = Dropout(0.5)(se1)
se3 = LSTM(256)(se2) #LSTM layer
decoder1 = add([fe2, se3])
decoder2 = Dense(256, activation='relu')(decoder1)
outputs = Dense(vocabulary_size, activation='softmax')(decoder2)
model = Model(inputs=[inputs1, inputs2], outputs=outputs)
model.summary()
```

```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_3 (InputLayer)            (None, 32)           0
__________________________________________________________________________________________________
input_2 (InputLayer)            (None, 2048)         0
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 32, 200)      330400      input_3[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 2048)         0           input_2[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 32, 200)      0           embedding_1[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 256)          524544      dropout_1[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 256)          467968      dropout_2[0][0]
__________________________________________________________________________________________________
add_1 (Add)                     (None, 256)          0           dense_1[0][0]
                                                                 lstm_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 256)          65792       add_1[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 1652)         424564      dense_2[0][0]
==================================================================================================
Total params: 1,813,268
Trainable params: 1,813,268
Non-trainable params: 0
__________________________________________________________________________________________________
```

Summary

```
* Embedding all the inputs
*. Using 'relu' activation
*. Here we are using LSTM layer
```

In [38]:

```
model.layers[2]
```

Out[38]:

```
<keras.layers.embeddings.Embedding at 0x7f46fba02f28>
```

In [39]:

```
model.layers[2].set_weights([word_matrix])
model.layers[2].trainable = False
model.compile(loss='categorical_crossentropy', optimizer='adam') #Adam optimizer
```

Summary

```
*. Using loss as categorical crossentropy
*. Using adam optimizer
```

In [40]:

```
epochs = 10
number_pics_per_bath = 3
steps = len(train_descriptions)//number_pics_per_bath
```

In [41]:

```
for i in range(epochs):
    generator = data_generator(train_descriptions, train_features, word_to_index, max_len, number_p
ics_per_bath)
    model.fit_generator(generator, epochs=1, steps_per_epoch=steps, verbose=1)
    #model.save('model_weights/model_1' + str(i) + '.h5')
    #model.save_weights('model_weights.h5')
    model.save_weights('model_weights.h5')
```

```
Epoch 1/1
2000/2000 [==============================] - 972s 486ms/step - loss: 4.1086
Epoch 1/1
2000/2000 [==============================] - 957s 478ms/step - loss: 3.4116
Epoch 1/1
2000/2000 [==============================] - 958s 479ms/step - loss: 3.1937
Epoch 1/1
2000/2000 [==============================] - 961s 480ms/step - loss: 3.0625
Epoch 1/1
2000/2000 [==============================] - 965s 482ms/step - loss: 2.9674
Epoch 1/1
2000/2000 [==============================] - 958s 479ms/step - loss: 2.8955
Epoch 1/1
2000/2000 [==============================] - 961s 481ms/step - loss: 2.8379
Epoch 1/1
2000/2000 [==============================] - 960s 480ms/step - loss: 2.7930
Epoch 1/1
2000/2000 [==============================] - 959s 479ms/step - loss: 2.7526
Epoch 1/1
2000/2000 [==============================] - 961s 481ms/step - loss: 2.7186
```

In [42]:

```
model.optimizer.lr = 0.0001
epochs = 10
number_pics_per_bath = 6
steps = len(train_descriptions)//number_pics_per_bath
```

In [43]:

```
for i in range(epochs):
    generator = data_generator(train_descriptions, train_features, word_to_index, max_len, number_p
ics_per_bath)
    model.fit_generator(generator, epochs=1, steps_per_epoch=steps, verbose=1)
    #model.save('model_weights/model_1' + str(i) + '.h5')
    #model.save_weights('model_weights.h5')
    model.save_weights('model_weights.h5')
```

```
Epoch 1/1
1000/1000 [==============================] - 724s 724ms/step - loss: 2.6413
Epoch 1/1
1000/1000 [==============================] - 717s 717ms/step - loss: 2.6157
Epoch 1/1
1000/1000 [==============================] - 718s 718ms/step - loss: 2.5957
Epoch 1/1
1000/1000 [==============================] - 717s 717ms/step - loss: 2.5778
Epoch 1/1
1000/1000 [==============================] - 719s 719ms/step - loss: 2.5611
Epoch 1/1
1000/1000 [==============================] - 720s 720ms/step - loss: 2.5399
Epoch 1/1
1000/1000 [==============================] - 723s 723ms/step - loss: 2.5222
Epoch 1/1
1000/1000 [==============================] - 715s 715ms/step - loss: 2.5060
Epoch 1/1
```

```
1000/1000 [==============================] - 717s 717ms/step - loss: 2.4904
Epoch 1/1
1000/1000 [==============================] - 721s 721ms/step - loss: 2.4800
```

In [44]:

```
model.load_weights('model_weights.h5') #loading the weights
```

In [45]:

```
images="image_data/Flicker8k_Dataset/"
test_images=load(open('image_data/encoded_test_images.pkl','rb'))
```

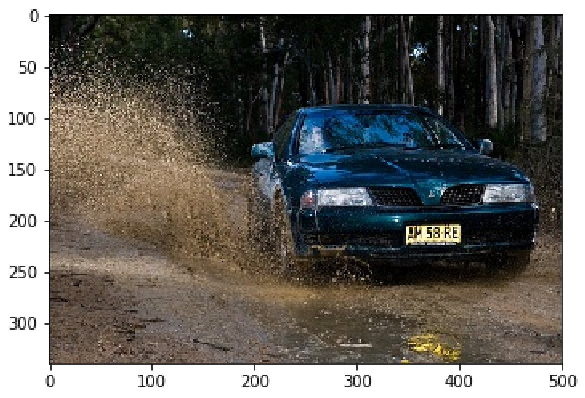
In [46]:

```
def greedySearch(photo):
    in_text = 'startseq'
    for i in range(max_len): # predicting  words max upto size of max length
        sequence = [word_to_index[w] for w in in_text.split() if w in word_to_index] # Finding the
index of words
        sequence = pad_sequences([sequence], maxlen=max_len) #padding the sequence of words upto ma
x length of description
        yhat = model.predict([photo,sequence], verbose=0) # predicting the next word from the input
ted padded sequence
        yhat = np.argmax(yhat)
        word = index_to_word[yhat] #Finding the index for the predicted word
        in_text += ' ' + word # adding the predicted word to the text sequence
        if word == 'endseq':
            break
    final = in_text.split()
    final = final[1:-1]
    final = ' '.join(final)
    return final
```

Test case :1

In [53]:

```
z=0
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



```
Greedy: car driving through mud puddle
```

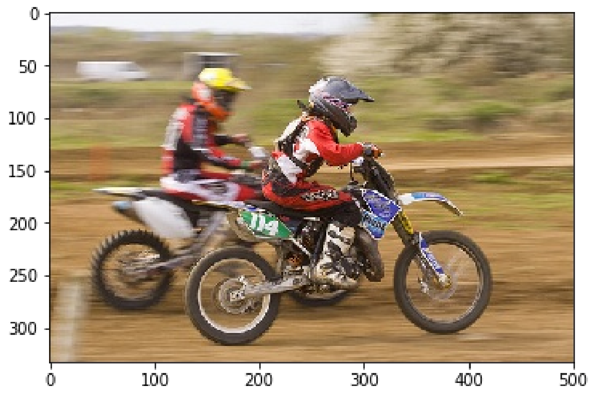
Output : Model predicted correctly

Test case :2

Test case :2

In [59]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



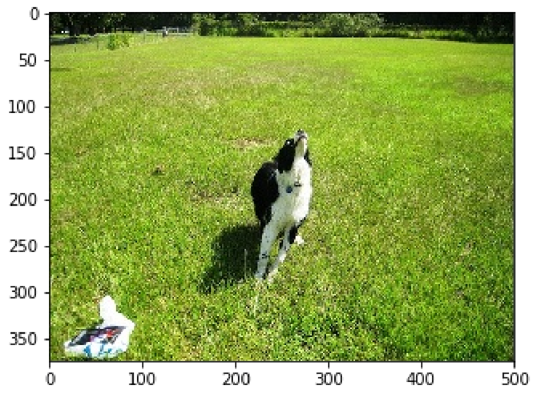
```
Greedy: motorcyclist is riding on racetrack
```

Output : Model predicted correctly

Test case :3

In [63]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



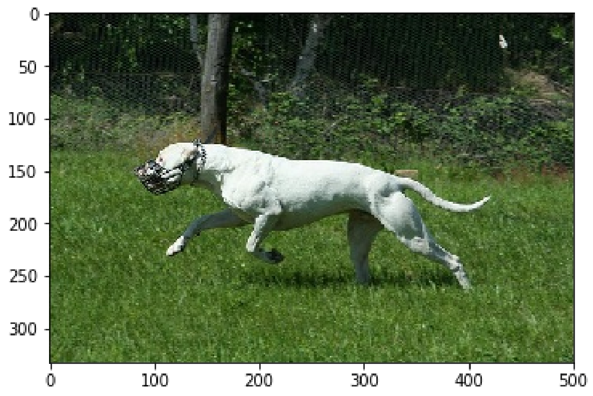
```
Greedy: dog is running through the grass
```

Output : Model predicted correctly

Test case :4

In [65]:

```
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



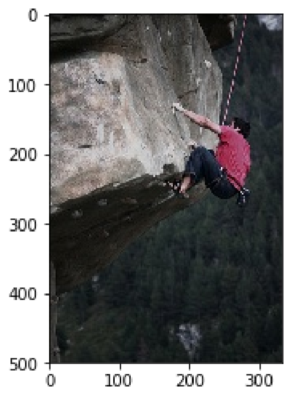
```
Greedy: dog is running on the grass
```

Output : Model predicted correctly

Test case :5

In [69]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



```
Greedy: man climbing sheer rock face
```

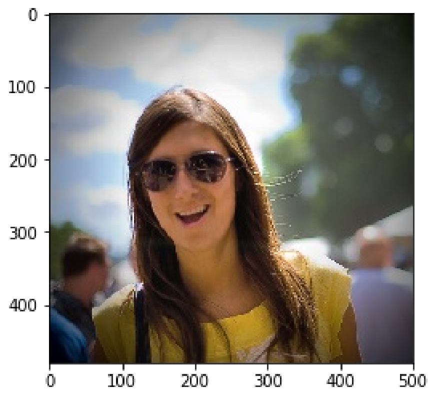
Output : Model predicted correctly

Test case :6

In [70]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
```

```
plt.show()
print("Greedy:",greedySearch(image))
```



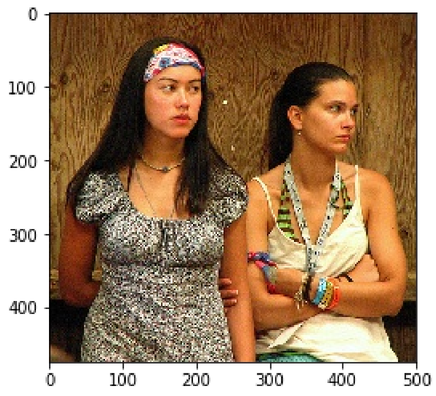
```
Greedy: woman with black hair and sunglasses smiles
```

Output : Model predicted correctly, but there is some semantic error

Test case :7

In [77]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



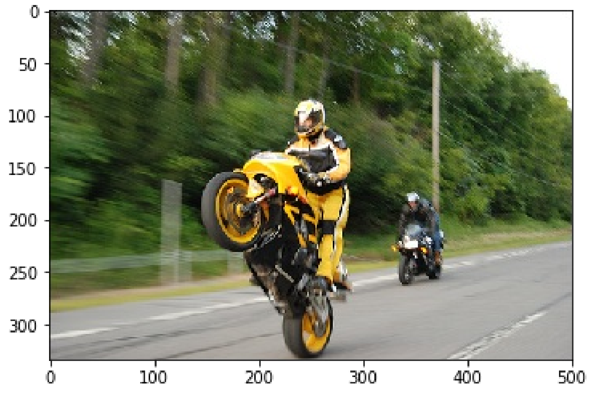
```
Greedy: two women in black posing for picture
```

Output : Model predicted correctly

Test case :8

In [79]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



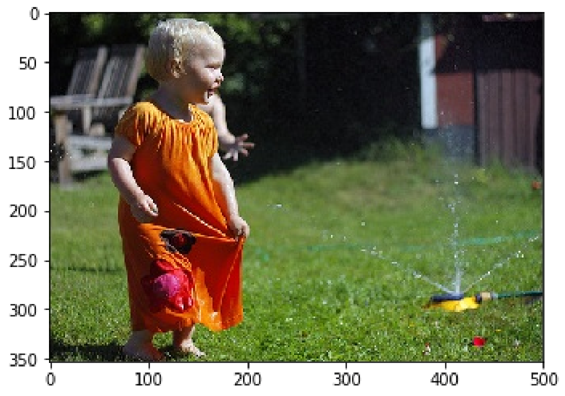
```
Greedy: man in yellow outfit riding motorcycle
```

Output : Model predicted correctly

Test case :9

In [82]:

```python
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



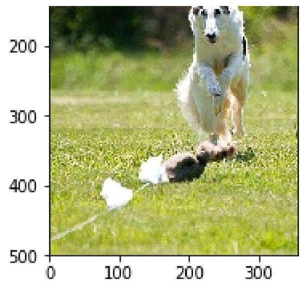
```
Greedy: young boy in blue shirt plays with toy
```

Output : Model predicted worse, colour of shirt is not predicted correctly

Test case :10

In [83]:

```python
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



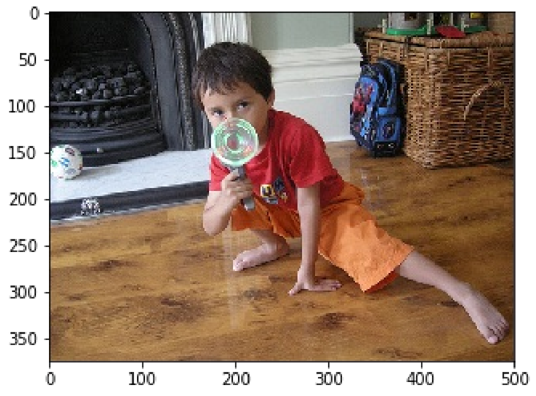
```
Greedy: dog is running through field
```

Output : Model predicted correctly

Test case :11

In [84]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



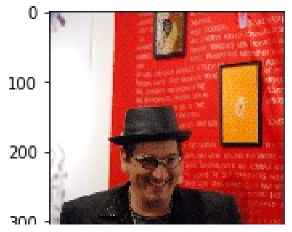
```
Greedy: man in blue shirt is sitting on the grass
```

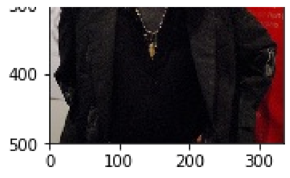
Output : Model predicted only one sitting position correctly, shirt colour and floor is predicted incorrectly

Test case :12

In [85]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```





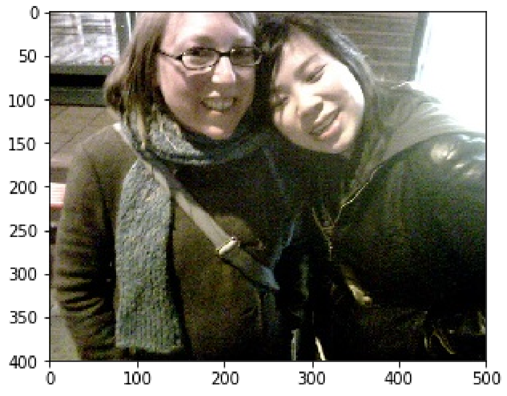
```
Greedy: man in black shirt and glasses is sitting on bench
```

Output : Model predicted correctly

Test case :13

In [86]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



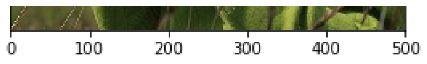
```
Greedy: woman with brown hair and man in black shirt smile for the camera
```

Output : Model predicted correctly

Test case :14

In [87]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



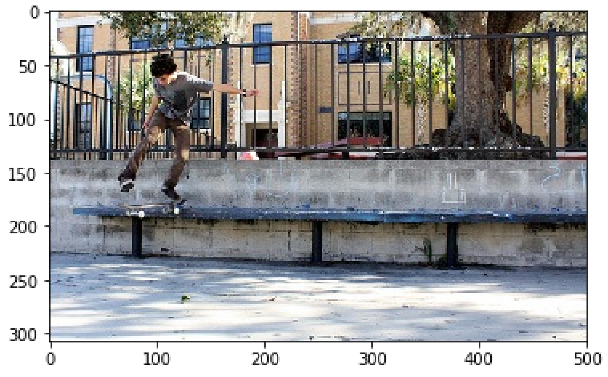
```
Greedy: young boy with blue shirt and blue shirt is holding football in his mouth
```

Output : May be model predicted football because of the shape in the image

Test case :15

In [88]:

```python
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



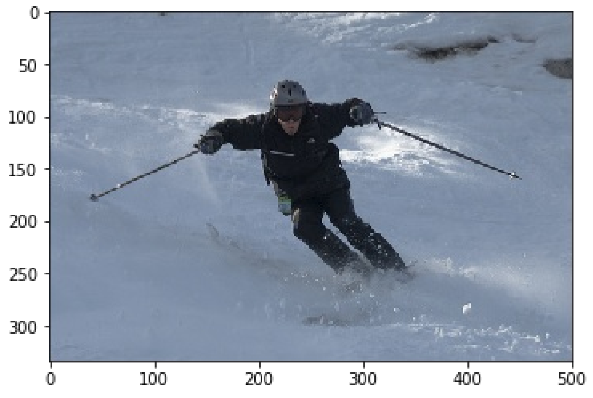
```
Greedy: skateboarder doing trick on ramp
```

Output : Model predicted correctly

Test case :16

In [90]:

```python
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



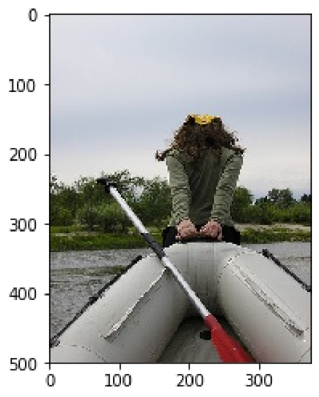
```
Greedy: man in red jacket is skiing on thin mountain
```

Output : Model predicted correctly,but the colour of the shirt is predicted wrong

Test case :17

In [91]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



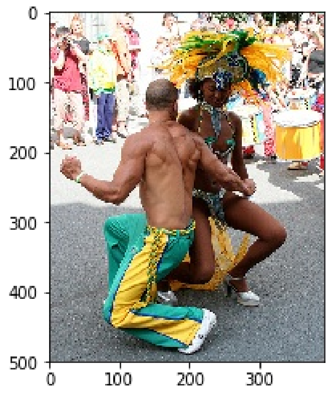
```
Greedy: the man is sitting on boat looking at the sunset
```

Output : Model predicted correctly

Test case :18

In [93]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



```
Greedy: man in black shirt is standing next to man in suit
```

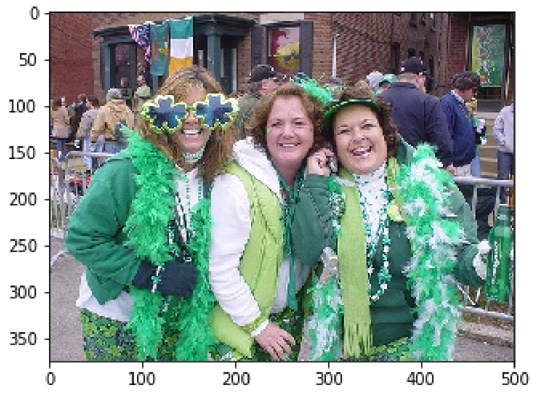
In [ ]:

```
#### Output : Model predicted worse
```

Test case :19

In [97]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



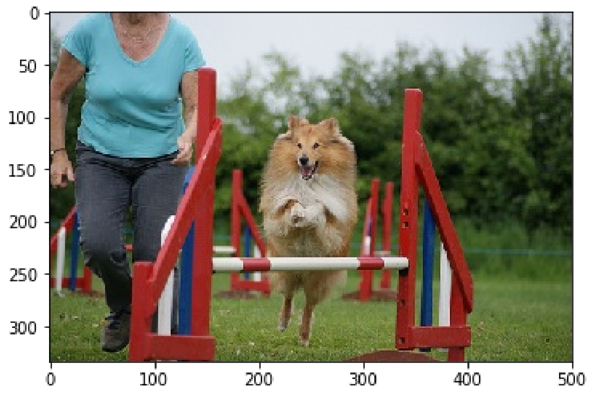
```
Greedy: two girls are smiling and laughing
```

Output : Model predicted partially correct, instead of three girls, model outputted only two

Test case :20

In [112]:

```
z+=1
pic = list(test_images.keys())[z]
image = test_images[pic].reshape((1,2048))
x=plt.imread(images+pic+'.jpg')
plt.imshow(x)
plt.show()
print("Greedy:",greedySearch(image))
```



```
Greedy: dog jumping over hurdle
```

Output : Model predicted correctly

Conclusion

```
    *. Loading the text data
    *. Cleaning the text data and saving the text data in a file
    *. Loading the Inception V3 bottleneck features
    *. For every image, we are loading the bottleneck features for that image
    *. Loading the glove models
    *. For every word, we are loading the 200 dimension glove vectors
    *. Taking max length of the descriptions
    *. We are taking every description
    *. Taking every word with padding upto maximum length of description and taking next
occurring word as target word.
    *. Instead of words, we are taking index of the words
    * Coverting all the target values to the categorical values
    * Training the model with LSTM layer
    *. In Greedy search, we are predicting the target index, we are outputting the word for
that index.
```

Reference

```
        1.https://machinelearningmastery.com/develop-a-deep-learning-caption-generation-mode
l-in-python/
        2.https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-d
escribe-pictures-c88a46a311b8
        3.https://github.com/hlamba28/Automatic-Image-
Captioning/blob/master/Automatic%20Image%20Captioning.ipynb
        4.https://www.youtube.com/watch?v=yk6XDFm3J2c
        5.https://arxiv.org/abs/1411.4555
        6.https://arxiv.org/abs/1703.09137
```